Multi-agent Reinforcement Learning for Planning and Conflict Resolution in a Dynamic Domain

Sachiyo Arai Katia Sycara
The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue Pittsburgh, PA 15213 USA
+1 (412) 268 7019
{sachiyo, katia}@cs.cmu.edu

1. Problem Domain and Approach

We present an approach known as Profit-sharing that allows agents to learn effective behaviors within dynamic and multi-agent environments, where the agents are competitive and may have to face resource conflicts, perceptual aliasing and uncertainty of other agents' intentions. A dynamic domain based on a *NEO (non-combatant evacuation operation)* is described.

1.1 Problem Domain

Non-combatant evacuation operations, or NEOs, have been used to test a variety of coordination strategies. Though real-world NEOs have many constraint and resource conflicts, the domain used in this study models multiple transportation vehicles which transfer groups of evacuees to safe shelters. Each transport is operated asynchronously by an autonomous agent, which makes its own decision based on locally available information.

The Neo domain consists of a grid world with multiple transporter agents, each of which carries a group of evacuees. The goal of a transporter agent is to ferry its group to one of the shelters as quickly as possible. However, there may be conflicts, as transporters cannot co-exist in the same location at the same time (Figure 1a). In addition, the location of the shelters changes over the time. In dynamic domains such as this, agents should exhibit reactive behaviors rather than deliberative ones. We claim that the only effective approach is to learn reactive behaviors through trial and error experiences, since it is very difficult to know in advance what effective action should be taken at each possible state of the environment. Each transporter agent is modeled as a reinforcement learning entity in an unknown environment, where there is no communication with the other agents, and there are no intermediate sub-goals for which intermediate rewards can be given. It should be noted that there are other agents within the environment that are also learning independently of each other, without sharing sensory inputs or policies. As a result, the other agents appear as additional components within the environment, whose behavior is dynamic and unpredictable.

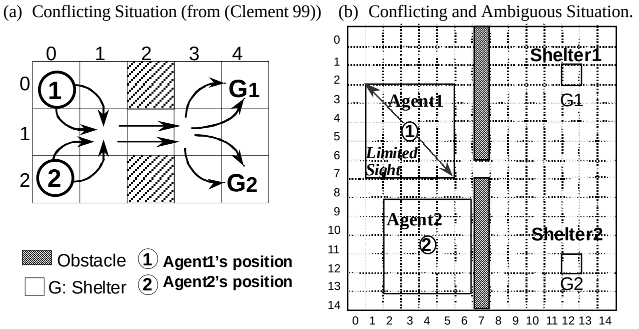
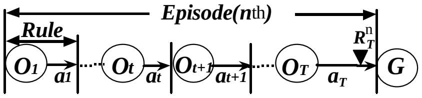

(a) Conflicting Situation (from (Clement 99))
(b) Conflicting and Ambiguous Situation.

Figure 1: Two Agents moving within the grid world. Figure(a) has been reproduced from [1].

1.2 Profit-sharing Approach

Our multi-agent reinforcement learning approach is based on Profit-sharing, originally proposed by [2]. The original version used Profit-sharing as a credit assignment method. However, this approach does not guarantee the rationality of an acquired policy. To guarantee convergence to a rational policy in a non-Markovian domain like *NEOs* which includes multiple learning entities, we introduce the *Rationality Theorem*[3](see Figure 2 Eq.1 and Eq.2). A rational policy is one that is guaranteed to converge on a solution; i.e. the agent should not become trapped within infinite loops in the state machine.



[Credit Assignment Method]

***Profit-sharing -our approach**[Miyazaki 94]

Theorem 1: Rationality Theorem[Miyazaki 94]

Any ineffective rule can be suppressed iff

$$\forall t = 1,2,3....,T.\ L\sum_{j=0}^{t} f(R,j) < f(R,t) \qquad \text{Eq.1}$$

$$W_{n+1}(O_t, a_t) \leftarrow W_n(O_t, a_t) + f(R_T^n, t) \qquad \text{Eq.2}$$

***Profit-sharing plan**[Grefenstette 88]

$$W_{n+1}(O_t, a_t) \leftarrow W_n(O_t, a_t) + \alpha(R_T^n - W_n(O_t, a_t)) \qquad \text{Eq.3}$$

***Q-learning**

$$Q_{n+1}(O_t, a_t) \leftarrow Q_n(O_t, a_t) + \alpha(r + V(O_{t+1}) - Q_n(O_t, a_t)) \qquad \text{Eq.4}$$

$$V(O_{t+1}) = \max_{b \in A_t} Q(O_{t+1}, b) \qquad \text{Eq.5}$$

R_T^n : Reward given at Goal in the n-th trial.
(Ot, at) : Observational State and action at time ***t***
W (Ot, at) : Weight of the
Observational state-action pair (***Ot, at***)
f : Reward assignment

Figure 2: Credit Assignment Methods.

2. Experiments

Two *NEO* grid worlds, as shown in Figure1, were designed to compare our Profit-sharing approach with Q-learning[4]. In both cases, two agents started from different locations, and their task was to learn policies for finding one of two shelters as quickly as possible. There are five actions within the action set, A_t=*{Stay, Up, Right, Down, Left}*. However, both agents cannot occupy the same position at the same time. In the grid-world of Figure1a, the number of location is small and the agents can see the whole environment. In the grid-world of Figure1b, the perceptual distance of each agent is only a 5×5 region; each agent see a shelter or the other agent when they are no more than two moves away.

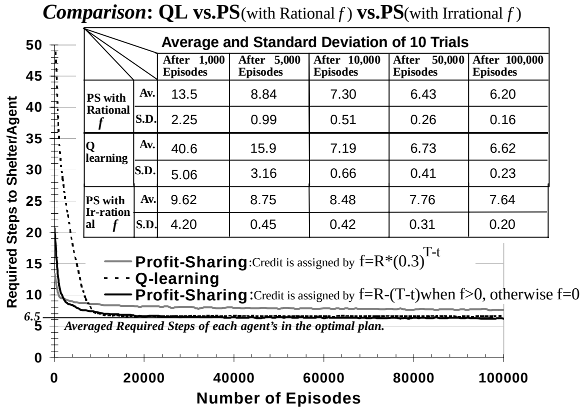
In each episode, the order in which the two agents move is determined randomly. Agent always start in the same location (i.e. (0, 0) & (0, 2) in the smaller world, and (0, 0) & (0, 14) in the larger one). The location of the shelters is determined by one of two experimental settings. In the first, their location is static. In the second, the location of the shelters varies within the right half of the grid world in each episode.

The learning parameters were selected as follows:

Profit-sharing: A geometrically decreasing function (common ratio=0.3) was used as a credit assignment function.

Q-learning: The learning rate α (=0.05) and discounting factor γ (= 0.9) in Eq.4 of Figure 2. When the agent reaches the goal state (i.e. the shelter), it receives a reward of 1.0. The Q-learning agent uses the Boltzmann distribution (T=0.2) to select its action.

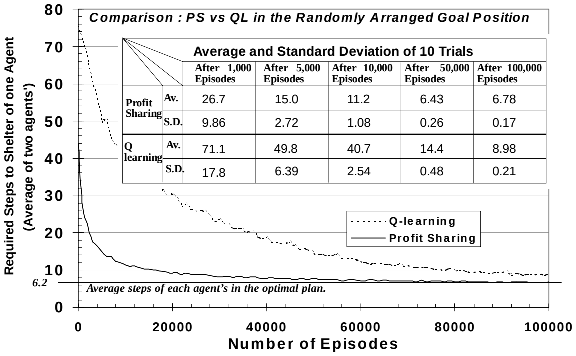
Figure 3 shows the results of the experiment where the location of the shelters was fixed for each episode. Figure 4 shows the results of the experiment where the location of the shelters varied in each episode. Figure 5 shows the results of the experiment where two grid were used ; the 15×15 world illustrated in Figure 1b, and similar but smaller 7×7 world. The results illustrated in Figure 5 indicate that Q-learning fails to converge for either world when the location of the shelter is varied. However, Q-learning performs well when the shelter location are fixed. This is not surprising, as Q-learning learns deterministic policies for Markov Decision Processes, and hence is unsuited for dynamic and uncertain domains. However, Profit-sharing collects stochastic data and reinforces useful rules using the *Rationality Theorem.*



Average and Standard Deviation of 10 Trials

		After 1,000 Episodes	After 5,000 Episodes	After 10,000 Episodes	After 50,000 Episodes	After 100,000 Episodes
PS with Rational *f*	Av.	13.5	8.84	7.30	6.43	6.20
	S.D.	2.25	0.99	0.51	0.26	0.16
Q learning	Av.	40.6	15.9	7.19	6.73	6.62
	S.D.	5.06	3.16	0.66	0.41	0.23
PS with Ir-rational *f*	Av.	9.62	8.75	8.48	7.76	7.64
	S.D.	4.20	0.45	0.42	0.31	0.20

Figure 3: Performance of the agent in the Conflicting Situation: Fixed Goal.



Average and Standard Deviation of 10 Trials

		After 1,000 Episodes	After 5,000 Episodes	After 10,000 Episodes	After 50,000 Episodes	After 100,000 Episodes
Profit Sharing	Av.	26.7	15.0	11.2	6.43	6.78
	S.D.	9.86	2.72	1.08	0.26	0.17
Q learning	Av.	71.1	49.8	40.7	14.4	8.98
	S.D.	17.8	6.39	2.54	0.48	0.21

Figure 4: Performance in the Conflicting Situation: Randomly Arranged Goals.

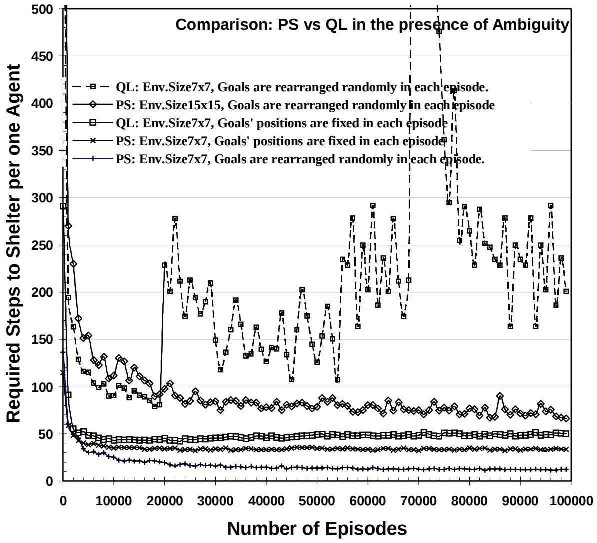


Figure 5: Performance in the Dynamic and Uncertain Domain.

3. ACKNOWLEDGMENTS

This research has been partially funded by Darpa contract F30602-98-2-0138, ONR Contract N00014-96-1222 and NSF grant IRI-9612131.

4. REFERENCES

[1] Clement, B. J., and Durfee, E. H. Top-Down Search for Coordinating the Hierarchical Plans of Multiple Agents. *In Proceedings of the 3rd International Conference on Autonomous Agents* (1999), 252-259.

[2] Grefenstette, J. Credit Assignment in Rule Discovery Systems Based on Genetic Algorithms, *Machine Learning Vol.3* (1988), 225-245.

[3] Miyazaki, K. and Kobayashi, S. On the Rationality of Profit Sharing in Partially Observable Markov Decision processes, *In Proceedings of the 5th International Conference on Information Systems Analysis and Cynthesis,* (1999), http://www.fe.dis.titech.ac.jp/~teru/papersj.html

[4] Watkins, C., and Dayan P. Technical note: Q-learning, *Machine Learning Vol.8* (1992), 55-68.